Vol. 00 no. 0 2006, pages
doi:10.1093/bioinformatics/bti283

ON-LINE SUPPLEMENT:

SPACER: Identification of *cis*-regulatory elements with non-contiguous critical residues

Arijit Chakravarty[1], Jonathan M. Carlson[2], Radhika S. Khetani[3], Charles E. DeZiel[3] and Robert H. Gross[3,*]

[1]Department of Cancer Pharmacology, Millennium Pharmaceuticals Inc., Cambridge, MA, [2]Department of Computer Science and Engineering, University of Washington, Seattle, WA and [3]Department of Biology, Dartmouth College, Hanover, NH

ABSTRACT

This supplement provides a detailed description of the SPACER algorithm, the experimental methods, datasets, and the results of performance comparisons for the algorithm.

1 OVERVIEW OF SPACER

SPACER is based on three efficient beam search algorithms combined in series. Beam search algorithms greedily explore a collection of most significant motifs, making a relaxed greedy assumption that the optimal motif is composed of relatively high scoring (but not necessarily the best) submotifs (Russell and Norvig, 1995).

In previous work, we described BEAM and PRISM (Carlson *et al.,* 2006a,b), aimed at the identification of short non-degenerate motifs and short degenerate motifs respectively. BEAM starts from the most over-represented short non-degenerate motifs and extends them to find the most over-represented non-degenerate motifs of any arbitrary length. PRISM starts with the top motifs BEAM outputs and 'generalizes' them, systematically permitting degeneracies at each base position in the motif to identify the most over-represented degenerate motifs of any arbitrary degree of degeneracy.

Neither BEAM nor PRISM is particularly well suited for the detection of long, highly degenerate motifs with non-contiguous critical residues, or bipartite motifs of the form A-S_N-B (where A and B are independent and separated by a spacer region S_N of N bases). BEAM, focused on the identification of over-represented core subsequences, will find the non-degenerate ends of bipartite motifs, but is unable to extend such motifs beyond the spacer region, which is not overrepresented. PRISM, when given one non-degenerate binding site, is likely to identify a motif that describes the *cis*-regulatory set well, but it depends on BEAM to provide the seed motif. Previously, we have shown that BEAM performs well at finding overrepresented motifs with low levels of degeneracy as long as their degree of overrepresentation is above a certain threshold (Carlson *et al.,* 2006a). Unfortunately, this threshold increases linearly with the degeneracy of the consensus motif that describes the *cis*-regulatory set, making this approach impractical for the discovery of gapped binding sites, as such sites would have to be very highly overrepresented in order to be detectable.

We therefore borrowed aspects of both BEAM and PRISM to create a motif-finding program aimed specifically at the identification of long highly degenerate motifs with non-contiguous critical residues (Supplementary figure 1). Our program, SPACER, uses as its objective function the Sig score proposed by van Helden, André and Collado-Vides (1998), which is a negative log-transformed *p*-value based on statistical over-representation that is corrected for multiple tests - in this case, the number of motifs of a given length and degeneracy.

The first stage of SPACER, referred to here as cSPACER, uses a modified version of BEAM to identify bipartite motifs that are statistically over-represented in the group of target upstream sequences (relative to all upstream regions in a given genome). For each of the most over-represented bipartite motifs, SPACER systematically explores the hamming space around the end regions A and B using PRISM. The result is a consensus sequence that encompasses the full range of possible binding sites.

Next, SPACER looks for weak base preferences in the spacer region S_N, using a recombination algorithm that starts with the observed instances of the motif in the upstream regions and systematically recombines those instances to retain combinations that improve the degree of over-representation of the original motif. The recombination algorithm in SPACER is specifically designed to start from a string of n's and identify weak base preferences within it. This algorithm is distinct from the hill-climbing algorithm used in PRISM, as it does not require lookups involving a string of n's (which are very slow, for details see section on runtime complexity).

Finally, for each degenerate bipartite motif, the putative binding sites predicted by SPACER are aligned to yield a position weight matrix.

2 METHODS

*To whom correspondence should be addressed.

2.1 Calculation of *Sig* score

SPACER uses the *Sig* score to measure the statistical overrepresentation of each motif (van Helden *et al.,* 1998). We previously described our modifications and justification for the *Sig* score (Carlson *et al*, 2006b). Briefly, the *Sig* score is the negative log of the Bonferroni-corrected Poisson probability that the motif occurs at least as often as it is observed. In motif-finding applications, the Bonferroni correction is an estimate of the number of potential motifs in the relevant motif category (as defined by the length and degree of degeneracy of the motif). This correction allows us to compare motifs of different length and degeneracy against each other. We estimate the number of motifs in each category by multiplying together the number of choices at each position (that is, the number of IUPAC symbols with the same degree of degeneracy as the symbol at that position),

$N(m) = \Pi_i \text{ Choose}(4, |b_i|)$,

where $|b_i|$ is the number of non-degenerate bases the i^{th} base expands to. Thus, the motif ATTTC has 4^5 motifs that are similar to it and NNNNN has one, precisely as we would expect. The Poisson distribution assumes that the probability of a motif occurring at a given position is independent of the motifs at any other position. In the case of highly repetitive motifs, this assumption will cause the *Sig* score to grossly overestimate the significance of the motif. Because highly repetitive motifs are rarely functional, we mask out such sequences in cSPACER.

Accurate computation of the *Sig* score requires an accurate count of the number of occurrences of a motif in both the regulon and the genome. The number of occurrences of non-bipartite motifs can be efficiently counted using a suffix array (for a review, see Gusfield, 1997). Degenerate, non-bipartite motifs are expanded to all possible instantiations (non-degenerate words that match the degenerate consensus sequence), and the counts of these instantiations are summed. This approach, however, breaks down in the case of bipartite motifs, where a series of N n's leads to 4^N instantiations. To count bipartite motifs, we therefore developed a variation of a wildcard string search algorithm (see Gusfield, 1997 for the original algorithm).

The algorithm works as follows: given a motif $m\_=\_m_1s_1m_2s_2\ldots s_km_k$, where s_i is a string of n's of defined length separating submotifs m_i and m_{i+1}, the locations of each m_i are located using the suffix array. The locations of m_i and m_{i+1} are then checked to ensure a spacing that is consistent with the length of s_i. Regions that match the form $m_1s_1m_2s_2\ldots s_km_k$ are hits for m.

2.2 Beam Search Algorithm (cSPACER)

The first stage of SPACER, referred to in this work as cSPACER (canonical SPACER), is a modified version of BEAM (Carlson *et al.,* 2006a, see Supplementary figure 1).

BEAM enumerates all non-degenerate 5-mers, then iteratively extends each of the 100 most over-represented motifs. When extending motifs, BEAM alternates between adding the next bases to the left and right of the core motif.

cSPACER, specifically aimed at the identification of motifs of the form A-S_N-B (where S_N is a spacer region of n's) first enumerates all gapped 4-mers, with gap lengths ranging from 2 to 15. When extending motifs, cSPACER tries adding the base to both the left side of *A* and the right side of *B* and greedily selects the extension with the higher score for further evaluation.

2.3 Hill-climbing Algorithm

In the second stage of SPACER, PRISM is employed to optimize the highest scoring motifs returned from cSPACER (see Supplementary Figure 1). PRISM uses an iterative hill-climbing algorithm that attempts to improve the score of a given motif m = A-S_N-B by systematically perturbing every base in A or B (Carlson *et al.,* 2006b). The base modifications permitted by PRISM are limited to modifications that retain the original base(s) as a subset of the degenerate possibilities (for example, an A can be modified to a W, a W can be modified to an H, but an A cannot be modified to a T or an S). PRISM iteratively optimizes each position until the score ceases to improve. By optimizing each position independently of the others, PRISM searches the space of all possible degeneracies of a given motif that is L bases long in $O(L^2)$ time instead of $O(7^L)$ time with the trade-off that optimality cannot be guaranteed, because all possible degenerate motifs are not enumerated and scored.

This tradeoff between optimality and complete enumeration is unavoidable for consensus-based programs. YMF chooses to guarantee optimality by enumerating all possible motifs, but limits the size of the alphabet and motif length to make this search feasible (Sinha and Tompa, 2002). RSAT similarly enumerates all possible motifs of restricted length and alphabet, and so can guarantee optimality for short, nondegenerate motifs, but heuristically explores the space of longer, degenerate motifs without any such optimality guarantees (van Helden *et al.,* 2000).

PRISM's informed heuristic search strategy, while not optimal, is efficient and can search a much larger search space, including the full IUPAC alphabet (Carlson *et al.,* 2006b). This approach outperformed the restricted optimal approach on non-bipartite motifs (Carlson *et al.,* 2006b).

2.4 Recombination Algorithm

In the final stage of SPACER, bipartite motifs of the form m = A-S_N-B, where A and B may be degenerate, are specialized (made less degenerate) to identify weak base preferences in the spacer region (see Supplementary Figure 1). Specialization is carried out by a simple recombination algorithm that greedily merges the observed instantiations of the motif.

For any motif m, let $I(m)$ be the set of all actual instances of m in the given regulon. For any pair of motifs m_1 and m_2, let MERGE (m_1, m_2) be the motif that results from combining the bases at each position of m_1 and m_2. For example, MERGE (ACGT, TCCT) = WCST. Finally, we define a partition $S\_=\_S_0, S_1, \ldots, S_N$ for motifs of length N such that $m \in S_i$ if m has i degenerate bases.

The recombination algorithm works as follows: for each partition S_i, generate m' = MERGE (m_1,m_2), for all $m_1, m_2 \in S_i$; remove m_1 and m_2 from the set and place m' into the appropriate partition $S_{j \geq i}$. The partitions are processed in order until the only motif left is all N's. The highest scoring motif encountered along the way is returned.

In each partition, only the $\__2$ highest scoring motifs are considered. This beam width is justified by the bound $Sig(m') \leq Sig(m_1) + \mathrm{Sig}(m_2)$, which implies that only the highest scoring motifs need be considered (Carlson, *et al.,* 2006b). Furthermore, if m_1 and m_2 differ by one mismatch, the bound tends to be tight, necessitating the partitions.

2.5 Runtime complexity

With the beam widths set to experimentally determined constants, the typical running time for the SPACER algorithm on yeast is approximately 30 seconds on a 2GHz processor with 1GB RAM. SPACER's runtime complexity is described by $O(t(Sig) + L^2 * t(Sig) + (L+N) * t(Sig))$, where $t(Sig)$ is the time it takes to compute the *Sig* score of a single bipartite motif, L is the average length of the motifs, and N is the average length of the spacer region.

Motif lookup

In this context, the efficient computation of *Sig* is critical. The runtime for the lookup of bipartite motifs $t(Sig)$ can be described in terms of the time to look up a motif $m=m_1 s_1 m_2 s_2 \ldots s_k m_k$ with e ambiguous bases (excluding N's in the SPACER regions) in a sequence S as $O(E_S(m))$, where $E_S(m)$ is the expected number of occurrences of all m_i in S.$E_S(m)$ thus grows linearly with the size of S (the set of all upstream sequences) and inversely with the size of m's components $m_1 \ldots m_k$. The expected running time for a regulon where the highest scoring motif is m, with average submotif length l and average spacer length N is thus $O(E_S(m)\ l+N))$.

cSPACER

The running time of the first stage of the algorithm scales linearly with respect to the motif lookup time and the length of motifs explored.

Hill-climbing algorithm

HC has time complexity $O(l^2 * t(\mathrm{Sig}))$, where l is the (average) length of the submotifs A and B.

Recombination algorithm

Running the recombination algorithm on all binding sites scales linearly with respect to motif length and lookup time, and quadratically with respect to the number of instantiations of each motif in the regulon has time complexity $O(\__3 * \mathrm{L} * C^2 * t(Sig))$, where $\__3$ is the number of motifs run on this stage and C is the number of occurrences of the motif in the regulon. The scaling properties are substantially improved by only running the highest $\_'_3$ scoring motifs in each partition on the recombination algorithm. This changes the asymptotic running time to $O(\__3 * N * \_'^2_3 * t(Sig))$.

2.6 Biological Datasets

We sought to create biological datasets that were as large as possible for the selected classes of transcription factors. The *S. cerevisiae* regulons consisted of the full set of Zn(II)2Cys6 regulons for which binding sites were available in the *Saccharomyces cerevisiae* protein database as of May 2004 (SCPD, Zhu and Zhang, 1999; http://rulai.cshl.edu/SCPD/). The SCPD database contains a number of groups of genes that are known to share functional, well-characterized binding sites. For the purpose of testing the algorithms, we assume that a group of genes bound by the same protein constitutes a regulon and that the reported binding sites accurately describe the only binding sites of that protein in that regulon. This is in accordance with other performance comparisons (Shinozaki, Akutsu and Maruyama, 2003; Sinha and Tompa, 2003). The prokaryotic datasets were obtained from the PRODORIC databases as the entire set of regulons from *B. subtilis* and *E. coli* as of May 2004. (Munch *et al.,* 2003; http://prodoric.tu-bs.de/).).

Finally, the regulons for the unknown transcription factors were identified from data generated by Lee *et al.,* (2002), using Chromatin Immunoprecipitation (ChIP). The regulons we chose contain genes that bind their regulators with a p-value of ≤ 0.001 or ≤ 0.005.

2.7 Accuracy

In the performance comparison assays, the motifs returned by each algorithm were compared against the known binding sites using Pevzner and Sze's (2000) _ score as a measure of Accuracy. The _ score (referred to in this paper as Accuracy), is a measure of the nucleotide-level goodness of fit between the actual instances of two sets of motifs m_1 and m_2 in the set of co-regulated upstream sequences. Accuracy can be defined as follows: let U be a unique numbering of all the bases in the upstream sequences of a given set, and $B_U(m) \subseteq U$ be the set of bases that are covered by actual instances of m in U. Then Accuracy is defined the ratio of the number of bases that both motifs identify to the total number of bases that either of the two motifs identify:

$$\Phi_U(m_1, m_2) = [B_U(m_1) \cap B_U(m_2)] / [B_U(m_1) \cup B_U(m_2)].$$

This metric therefore takes both false positives and false negatives into account at the level of the individual bases that are actually covered by the motif, making it a good unbiased estimator of the Accuracy of the prediction against the known binding sites (regardless of the motif model used). In general, an Accuracy score of at least 0.2 represents a close match to the known motifs (as can be seen from the sequence logos in Supplementary Figure 2, for instance). The Accuracy for a program on a given regulon is scored for the top three motifs that the program reports and the motif with the highest Accuracy is reported. The same criteria were used for all other programs tested, and are in accordance with previous performance comparison studies (Shinozaki, Akutsu and Maruyama, 2003, Sinha and Tompa, 2003). For PWM-based models, where the threshold is needed to determine which putative binding sites match the model, we used the default output of each program, which included a list of matching binding sites.

3 RESULTS

3.1 Performance on Biological Datasets

We tested the full SPACER algorithm on *S. cerevisiae* regulons corresponding to transcription factors in the Zn(II)2Cys6 family (binding sites as well as regulated genes were obtained from the SCPD database; Zhu and Zhang, 1999). This constituted a more realistic test set for our algorithm, because, unlike the synthetic motifs used to develop the algorithm, known binding sites may overlap each other and vary in lengths and position-specific base preferences.

SPACER's results for each well characterized Zn(II)2Cys6 domain are shown in Supplementary Figure 2 (Accuracy provided in brackets). Sequence logos were created by aligning the sites that matched the consensus reported by the algorithm and using WebLogo (http://weblogo.berkeley.edu) to generate the logos. In all cases, the sites predicted by SPACER match the experimental sites recorded in SCPD by Accuracy ≥ 0.25, the expected score under the event where SPACER reports a motif that is exactly one base too long or too short.

In addition to validating SPACER on datasets with published binding sites, we used SPACER to predict binding sites for two previously uncharacterized transcription factors belonging to the Zn(II)2Cys6 family of DNA-binding proteins (Todd and Adrianopoulos, 1997). One of the transcriptions factors, MAL13 is involved in maltose fermentation, and was originally identified as a sequence homolog of MAL63 (Charron, Dubin and Michels 1986). The other is MAL33/MAL3R, a close sequence homolog of MAL13 that also plays a role in Maltose fermentation. *Sig* scores for the top predicted motifs for these two regulons were 120 and 307 respectively, well above the background *Sig* scores for the same number of genes randomly chosen from the genome (0.02–0.10). The top predicted motif for each of these regulons is listed in Supplementary Figure 2.

Although SPACER was modeled on the Zn(II)2Cys6 DNA binding motif, it is equally suited for the identification of prokaryotic motifs, which are often long and highly degenerate. Without changing any parameters from those determined on synthetic (yeast-derived) datasets, we ran SPACER on 39 different regulons from *S. cerevisiae*, *B. subtilis* and *E. coli*. The average Accuracy of the sites SPACER predicted was 0.24 (95% CI: 0.18–0.32) on this dataset. In 29 of the 39 cases, SPACER's Accuracy was at least 0.10 (Table 1). Qualitatively, the profiles of SPACER's predictions are a good visual match for the known binding sites (Supplementary Figure 2. Note that the motifs shown are the best results for *B. subtilis* and a range of results for *S. cerevisiae*. This is justified on the basis that the *S. cerevisiae* motifs are known to be bipartite, while the *B. subtilis* motifs are not necessarily bipartite).

When tested on this dataset, SPACER was found to outperform both BEAM and PRISM by a substantial margin. The average Accuracy for BEAM (0.13) and PRISM (0.14) was substantially lower than that of SPACER (0.24; $p<0.001$ for both comparisons against SPACER by a two-tailed paired t-test). SPACER won 84% of all head-to-head comparisons where the difference between two programs was at least 0.10.

We also looked at the Specificity (fraction of predicted nucleotides that overlap with published) and Sensitivity (fraction of published nucleotides that overlap with predicted) for each algorithm (Supplementary Figure 3). SPACER is about as specific as BEAM, with a Sensitivity that improves on that of PRISM.

SPACER is a general-purpose algorithm, focused on identifying motifs of arbitrary length and degree of degeneracy. Nonetheless, we note that SPACER (average Accuracy of 0.24) did not show a clear margin of superiority over cSPACER on this dataset (average Accuracy of 0.25). Thus, SPACER's competitive advantage on this dataset is provided by the cSPACER algorithm (which is focused on motifs of the form A-S_N-B, where A and B are binding regions separated by a space region S_N). For the *S. cerevisiae* regulons, this result can be explained simply by the canonical nature of these bipartite motifs. For *B. subtilis* and *E. coli*, cSPACER's strong performance was somewhat unexpected, though it is consistent with earlier reports that some bacterial *cis*-regulatory sets are essentially bipartite motifs (Robison, McGuire and Church *et al.,* 1998).

3.2 Performance comparison against other motif finders

To place SPACER's performance in a broader context, we ran ten other programs on the same datasets (Supplementary Table 2) We specifically chose programs that are specialized for bipartite motifs and that represent a range a motif representations and search strategies, including two Gibbs sampling algorithm using PWMs (Bioprospector, Liu, Brutlag and Liu, 2001; SesiMCMC, Favorov *et al.,* 2005), an Entropy Minimization algorithm using PWMs (Bipad, Bi and Rogan 2004), a k-mismatch algorithm aimed at prokaryotic bipartite motifs (Mitra, Eskin and Pevzner 2002) and two exhaustive consensus motif algorithms (RSAT, van Helden, Rios, Collado-Vides 2000; YMF, Sinha and Tompa 2002). We also compared general motif finders that are often used for all motif classes and represent a range of search strategies, including Gibbs sampling (AlignACE, Roth *et al.,* 1998; Gibbs, Thijs *et al.,* 2002), Expectation Maximization (MEME, Bailey and Elkan 1995) and bounded enumeration over mismatch consensus motifs (Weeder, Pavesi, Mauri and Pesole, 2001).

While previous performance comparisons painstakingly sought to optimize each program, we chose to run the programs as a biologist would, leaving unknowable parameters (such as expected motif length and coverage) at their default values. On principle, the only variable we changed was the species used for the background model and the number of motifs returned. SPACER allows a fair comparison in this way because it has no parameters besides internal ones which were permanently set based on synthetic datasets unrelated to the datasets in question. Weeder, the winner of Tompa *et al.*'s (2005) recent expert-user performance comparison, was run only on the Zn(II)2Cys6 regulons, as background sequences for the bacterial organisms were not available on WeederWeb (Pavesi *et al.,* 2004).

We used the evaluation criteria originally put forth by Sinha and Tompa (2003). In head-to-head comparisons, SPACER outperformed all 10 programs by a number of criteria (Supplementary Table 1). SPACER's average Accuracy (0.24) was between 10% and 250% higher

than each of the other programs. SPACER returned the highest score in more regulons (12) than any other program (MEME and RSAT are the next highest, with 6). Following Sinha and Tompa (2003), we looked at the frequency of clear wins in head to head comparisons, defined as an instance where one program outperformed the other program by a margin of at least 0.10. In such instances, SPACER had the higher score 78% of the time. RSAT had the second highest average Accuracy (0.22), but SPACER had the higher score in 13 of 18 (72%) of clear differences.

Breaking up the performance by species, the three consensus algorithms (SPACER, RSAT and YMF) score better than all the PWM and mismatch algorithms on *S. cerevisiae* (Supplementary Figure 4), reflecting the simple structure of the Zn(II)2Cys6 binding sites, which are well-described by simple consensus motifs. On *B. subtilis*, SPACER outperforms all other programs. Many programs perform similarly for *E. coli*, though again SPACER is among the top three performers.

A common criticism of consensus-based algorithms is that the representational constraints of the motif model make it difficult to achieve high Sensitivity. PWM-based algorithms, in turn, have been found to suffer from low Specificity in their predictions (Tompa *et al.,* 2005; Carlson *et al.,* 2006b), likely stemming from the representational limitations of their motif model (see, for example, Benitez-Bellon, Moreno-Hagelsieb and Collado-Vides 2002; King and Roth 2003; Gershenzon, Stormo and Ioshikhes 2005). Our performance comparisons are consistent with these observations: after SPACER, RSAT and YMF have the highest Specificity (offset by low Sensitivity), while AlignACE has the highest Sensitivity (offset by lower Specificity; Supplementary Figure 5). The reader may note that the prediction from any PWM method may be made more specific simply by changing the threshold at which a DNA sequence is considered to match the PWM. Any increase in Specificity will, however, be accompanied by a corresponding decrease in Sensitivity (see, for example, Benitez-Bellon et al 2002). For this study, we used the threshold specified by each program. While many of the other programs demonstrate clear tradeoffs between Specificity and Sensitivity, SPACER's Sensitivity and Specificity are roughly equal and both scores place it in the top two of the eleven programs tested (Supplementary Figure 4).

3.3 Performance in the presence of noise

The test sets we used were based on well characterized regulons, where every binding site is based on direct experimental evidence. In practice, putative regulons are often identified by microarray clustering of expression data. One study assessed the fraction of coregulated genes in a microarray cluster at 28% (Yeung, Medvedovic and Bumgarner 2004). Thus, to assess the performance of the different programs under noisy conditions, we introduced randomly selected genes from the *S. cerevisiae* genome into the 5 Zn(II)2Cys6 regulons. In each regulon, we introduced between 1 and 4 times as many randomly selected genes as the number of genes originally present in the regulon. The performance of all 6 programs decreased steadily in the face of increasing quantities of noise (Supplementary Figure 6). Predictably, such a proportional decrease in Accuracy across the different programs results in a greater number of situations where SPACER clearly outperforms the other programs (93% of head-to-head wins, Supplementary Table 1). Supplementary Figure 6 shows the effect on the average Accuracy of each program at different levels of noise. At each level of noise, SPACER's performance is superior to the other programs tested.

REFERENCES

Bailey,T.L. and Elkan,C. (1995) Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learn.*, **21**, 51–80.

Benitez-Bellon, E., Moreno-Hagelsieb, G. and Collado-Vides, J. (2002) Evaluation of thresholds for the detection of binding sites for regulatory proteins in *Escherichia coli* K12 DNA. *Genome Biol.*, **3**, research0013.

Bi,C. and Rogan,P.K. (2004) Bipartite pattern discovery by entropy minimization-based multiple local alignments. *Nucleic Acids Res.*, **32**, 4979–4991.Brazma,A., Jonassen,I., Eidhammer,I. and Gilbert,D. (1998) Approaches to the automatic discovery of patterns in biosequences. *J. Comput. Biol.*, **5**, 279–305.

Carlson,J.M., Chakravarty,A. and Gross,R.H. (2006a) BEAM: A beam search algorithm for the identification of *cis*-regulatory elements in groups of genes. *J. Comput. Biol.*, **13**, 686–701.

Carlson,J.M., Chakravarty,A., Khetani,R.S. and Gross,R.H. (2006b) Bounded search for *de novo* identification of degenerate *cis*-regulatory elements. *BMC Bioinformatics*, **7**, 254.

Charron,M.J., Dubin,R.A. and Michels,C.A. (1986) Structural and functional analysis of the MAL1 locus of *Saccharomyces cerevisiae*. *Mol. Cell. Biol.,* **6**, 3891-3899.

Crooks,G.E., Hon,G., Chandonia,J.M., Brenner,S.E. (2004) WebLogo: A sequence logo generator. *Genome Res.*, **14**,1188-1190.

Eskin,E. and Pevzner,P.A. (2002) Finding composite regulatory patterns in DNA sequences. *Bioinformatics*, **18 Suppl 1**, S354–S363.

Favorov,A.V., Gelfand,M.S., Gerasimova,A.V., Ravcheev,D.A., Mironov,A.A., Makeev,V.J. (2005) A Gibbs sampler for identification of symmetrically structured, spaced DNA motifs with improved estimation of the signal length. *Bioinformatics*, **21**, 2240–2245.

Gershenzon,N.I., Stormo,G.D. and Ioshikhes,I.P. (2005) Computational technique for improvement of the position-weight matrices for the DNA/protein binding sites. *Nucleic Acids Res.*, **33**, 2290-2301.

Gusfield,D (1997) Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press.

King,O.D. and Roth,F.P. (2003) A non-parametric model for transcription factor binding sites. *Nucleic Acids Res.*, **31**, e116.

Lee,T.I., Rinaldi,N.J., Robert,F., Odom,D.T., Bar-Joseph,Z., Gerber,G.K., Hannett,N.M., Harbison,C.T., Thompson,C.M., Fraenkel,E., Gifford,D.K., Young,R.A. (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, **298**, 799–804.

Liu,X., Brutlag,D.L. and Liu,J.S. (2001) BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac. Symp. Biocomput.*, **6**, 127–138.

Munch,R., Hiller,K., Barg,H., Heldt,D., Linz,S., Wingender,E. and Jahn,D. (2003) PRODORIC: prokaryotic database of gene regulation. *Nucleic Acids Res.*, **31**, 266-269.

Pavesi,G., Mauri,G. and Pesole,G. (2001) An algorithm for finding signals of unknown length in DNA sequences. *Bioinformatics* **17**, S206–S214.

Pavesi,G., Mereghetti,P., Mauri,G. and Pesole,G. (2004) Weeder Web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes. *Nucleic Acids Res*, **32** (Web Server issue), W199-W203.

Pevzner,P. and Sze,S.H. (2000) Combinatorial approaches to finding subtle signals in DNA sequences. *Proc. Int. Conf. Intell. Syst. mol. Biol.*, pp. 269–278. AAAI Press, San Diego, CA.

Robison,K., McGuire,A.M. and Church,G.M. (1998) A comprehensive library of DNA-binding site matrices for 55 proteins applied to the complete *Escherichia coli* K-12 genome. *J. Mol. Biol.*, **284**, 241–254.

Roth,F.P., Hughes,J.D., Estep,P.W. and Church,G.M. (1998) Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat. Biotechnol.* **16**, 939–945.

Russell,S. and Norvig,P. (1995) Artificial Intelligence: A Modern Approach. Prentice Hall. Schneider,T.D., Stephens,R.M.. (1990) Sequence Logos: A New Way to Display Consensus Sequences. *Nucleic Acids Res.*, **18**, 6097–6100

Schneider,T.D. and Stephens,R.M.. (1990) Sequence Logos: A New Way to Display Consensus Sequences. *Nucleic Acids Res.*, **18**, 6097–6100Shinozaki,D., Akutsu,T. and Maruyama,O. (2003) Finding optimal degenerate patterns in DNA sequences. *Bioinformatics*, **19 Suppl 2**, ii206-ii214.

Shinozaki,D., Akutsu,T. and Maruyama,O. (2003) Finding optimal degenerate patterns in DNA sequences. *Bioinformatics*, **19 Suppl 2**, ii206-ii214.

Sinha,S. and Tompa,M. (2002) Discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Res.*, **30**, 5549–5560.

Sinha,S. and Tompa,M. (2003) Performance comparison of algorithms for finding transcription factor binding sites. In *IEEE Symp. Bioinform. Bioeng.*, pp. 214–220.

Thijs,G., Marchal,K., Lescot,M., Rombauts,S., De Moor,B., Rouze,P. and Moreau,Y. (2002) A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *J. Comput. Biol.*, **9**, 447–464.

Todd,R.B. and Andrianopoulos,A. (1997) Evolution of a fungal regulatory gene family: the Zn(II)2Cys6 binuclear cluster DNA binding motif. *Fungal Genet. Biol.*, **21**, 388–405.

Tompa,M., Li,N., Bailey,T.L., Church,G.M., De Moor,B., Eskin,E., Favorov,A.V., Frith,M.C., Fu,Y., Kent,W.J., Ma-keev,V.J., Mironov,A.A., Noble,W.S., Pavesi,G., Pesole,G., Regnier,M., Simonis,N., Sinha,S., Thijs,G., van Helden,J., Vandenbogaert,M., Weng,Z., Workman,C., Ye,C., Zhu,Z. (2005) Assessing computational tools for the discovery of transcription factor binding sites. *Nat. Biotechnol.*, **23**, 137–144.

van Helden,J., André,B. and Collado-Vides,J. (1998) Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J. Mol. Biol.*, **281**, 827–842.

van Helden,J., Rios,A. and Collado-Vides,J. (2000) Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Res.*, **28**, 1808–1818.

Wasserman,W.W. and Sandelin,A. (2004) Applied bioinformatics for the identification of regulatory elements. *Nat. Rev. Genet.*, **5**, 276–287.

Yeung,K.Y., Medvedovic,M. and Bumgarner,R.E. (2004) From co-expression to co-regulation: how many microarray experiments do we need? *Genome Biol.*, **5**, R48.

Zhu,J. and Zhang,M.Q. (1999) SCPD: a promoter database of the yeast *Saccharomyces cerevisiae. Bioinformatics*, **15**, 607-611. http://rulai.cshl.edu/SCPD/

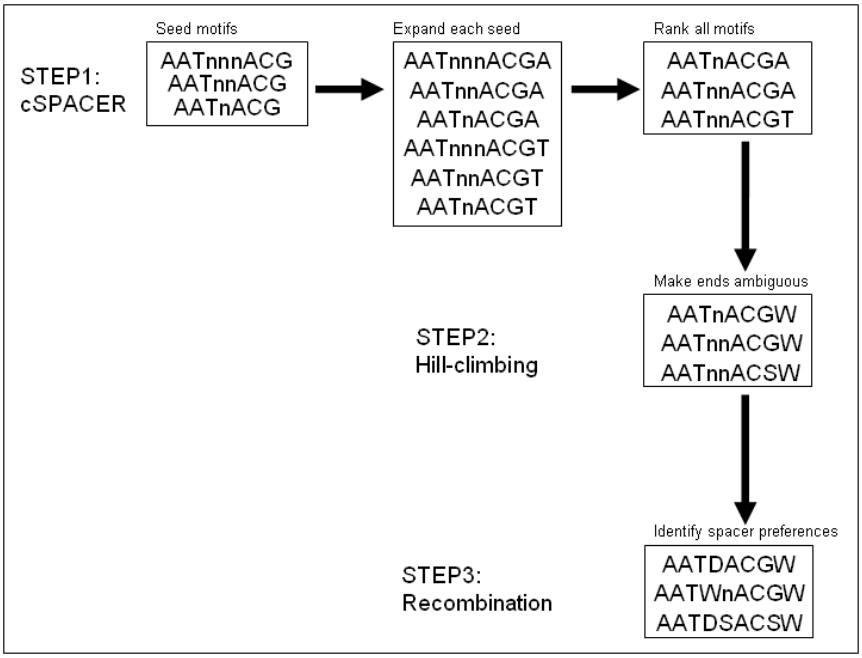


Figure 1: Schematic of SPACER and its component algorithms. Step 1: cSPACER enumerates all canonical bipartite motifs of the form A-S_N-B where A and B are 3bp long. It then iteratively expands the highest scoring motifs by adding bases to each end. Step 2: The 100 highest scoring motifs from cSPACER are passed to PRISM, which greedily generalizes the ends by trying more general IUPAC symbols. Step 3: SPACER then specializes the spacer region by iteratively combining the actual putative binding sites in the upstream region that match the consensus motif.

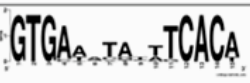
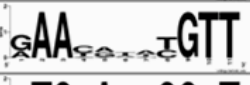
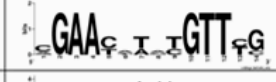
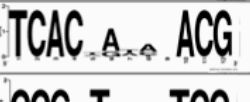
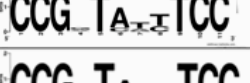
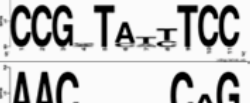
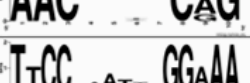
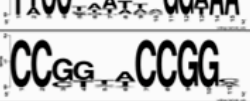
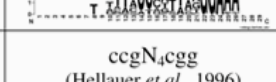
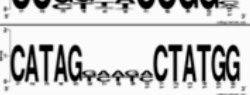
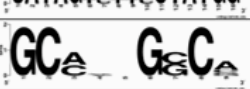
Regulon	Sequence logo reported by SPACER	Sequence logo from published binding sites
Fnr (1.00)		
XylR (0.67)		
LexA (0.61)		
CcpA (0.47)		
AraR (0.44)		
ABF1 (0.29)		
GAL4 (0.20)		
HAP1 (0.25)		
HSE (0.09)		
MCM1 (0.41)		
LEU3		ccgN_4cgg (Hellauer *et al.*, 1996)
MAL13		*Unknown*
MAL33		*Unknown*

Figure 2. Comparison of reported and actual sequence logos. The first five regulons are the five highest scoring predictions from *B. subtilis*, the rest are from *S. cerevisiae*, chosen to display a range of accuracies. Numbers in parentheses indicate prediction Accuracy, measured as the overlap between the predicted and known binding sites. PWMs were constructed from the instances of the motifs SPACER reported. Sequence logos were created from the PWMs using WebLogo (Crooks et al., 2004; Schneider and Stevens, 1990). X axis denotes position within the binding site. Y axis denotes information content in bits. The height of each letter represents the prevalence of that base at that position.

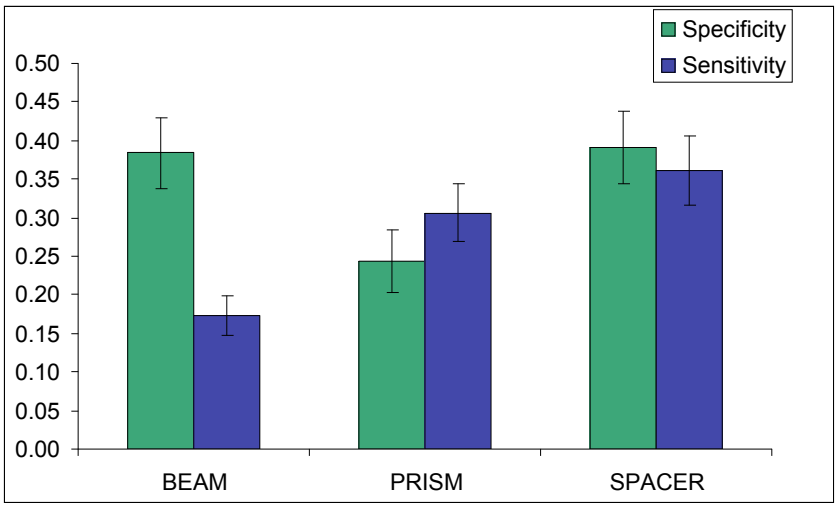


Figure 3. Specificity and Sensitivity for BEAM, PRISM and SPACER on the full dataset. Bars represent standard error. Specificity is defined as the proportion predicted binding site bases that overlap with known binding site bases. Sensitivity is defined as the proportion of known binding site bases that overlap with predicted binding site bases.

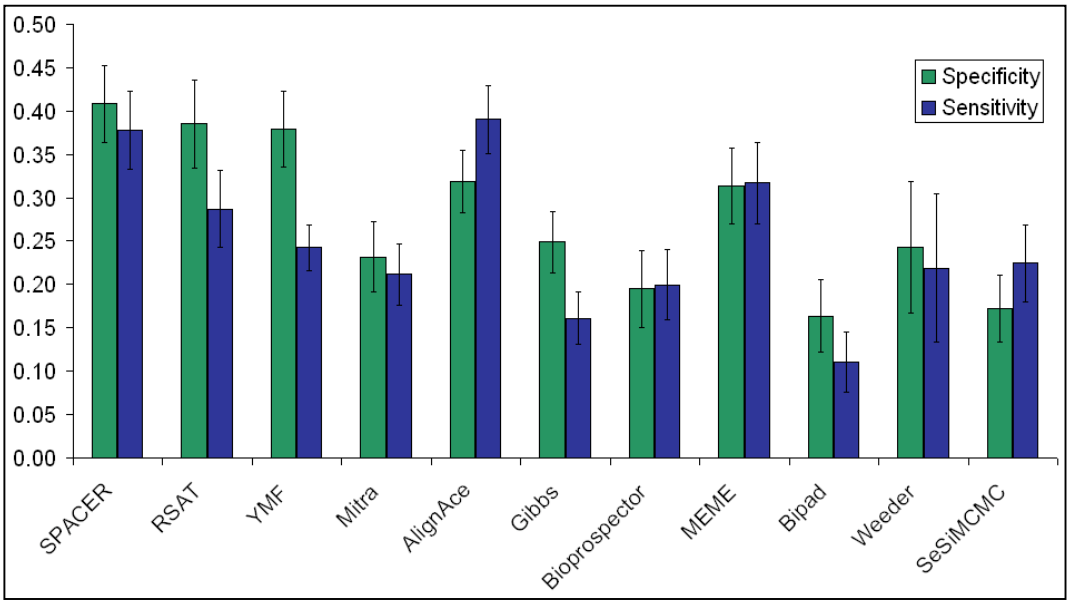


Figure 4. Average and standard error of Specificity and Sensitivity for each program. Specificity is defined as the proportion of predicted binding site bases that overlap with known binding site bases. Sensitivity is defined as the proportion of known binding site bases that overlap with predicted binding site bases.

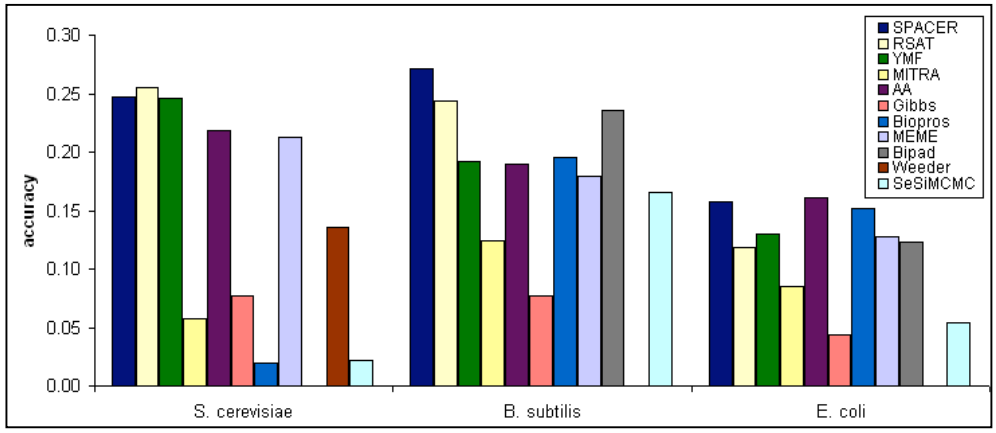


Figure 5. Average Accuracy of each motif finder on each organism. (Background distributions for *B. subtilis* and *E. coli* were not available for Weeder.)

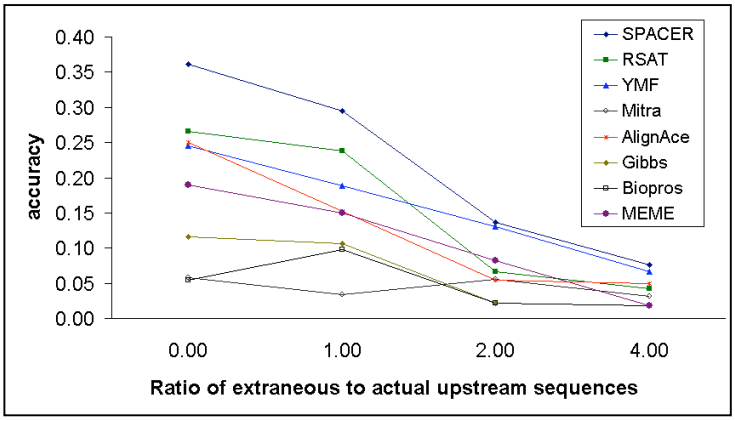


Figure 6. Average performance for motif finders as a function of noise over the *S. cerevisiae* data set. The x-axis indicates the number of randomly selected upstream sequences relative to the true size of the regulon.

Table 1. Summary of Accuracy scores on well-characterized regulons from *S. cerevisiae*, *B. subtilis*, and *E. coli*.

	SPACER	RSAT	YMF	MITRA	AA	Gibbs	Biopros	MEME	Bipad[a]	Weeder	SeSiMC
average	0.24	0.22	0.18	0.11	0.19	0.07	0.14	0.17	0.13	0.14	0.11
95% CI	0.06	0.07	0.05	0.04	0.05	0.03	0.06	0.05	0.07	0.12	0.05
wins[b]	12	6	1	3	4	0	2	5	2	0	6
# scores ≥ 0.50	3	3	2	0	3	0	2	3	1	0	2
# scores ≥ 0.33	14	8	6	2	7	1	3	7	3	1	4
# scores ≥ 0.10	29	25	27	15	30	12	12	22	8	3	15
# missing[c]	0	5	3	3	0	0	16	0	23	3	2
clear win for SPACER[d]	-	13	14	18	11	25	12	14	9	4	20
clear loss for SPACER[d]	-	5	5	3	6	1	6	5	2	1	4

[a] Bipad could not be configured to return more than one motif, so head to head comparisons were computed against SPACER's top Sig-scoring motif. For comparison, SPACER's average for the top motif on the regulons is 0.19. [b] A program "wins" a regulon if it has the highest Accuracy and that score is at least 0.10. [c] Number of regulons for which the program failed to return any results. These regulons were not included in the average and clear win/loss counts for those programs. We ran Weeder only on the *S. cerevisiae* regulons as the program does not have background distributions defined for the other species. [d] A clear win/loss is a difference of at least 0.10 between two programs on a regulon.

Table 2. Complete listing of Accuracy scores obtained by all programs. A '-' indicates that the program did not return a motif for this regulon. (For such regulons, Accuracy and Sensitivity are undefined and are not included in summary statistics, while Specificity is 0 by definition.) A score of n/a indicates the program does not have background distributions available for those organisms. These entries were not used for any summary statistics.

		SPACER	RSAT	YMF	MITRA	AA	Gibbs	Biopros	MEME	Bipad	Weeder	SeSiMCMC
S. cerevisiae	ABF1	0.29	0.36	0.35	0.01	0.28	0.00	0.01	0.00	0.00	0.03	0.00
	GAL4	0.20	0.21	0.20	0.11	0.18	0.11	0.01	0.13	0.00	0.10	0.14
	HAP1	0.25	0.20	0.13	0.00	0.14	0.07	0.00	0.22	0.00	0.06	0.00
	HSE1	0.09	0.44	0.38	0.11	0.12	0.12	0.08	0.27	0.00	0.37	-
	LEU3	0.07	0.11	-	-	0.08	0.06	0.00	0.07	0.00	-	0.00
	MCM1	0.41	0.21	0.17	0.06	0.53	0.12	0.04	0.33	0.00	0.12	0.01
	PUT3	0.34	0.24	-	-	0.11	0.00	-	0.43	-	-	0.00
	UME6	0.33	0.27	-	-	0.31	0.14	0.00	0.25	0.00	-	0.00
B. subtilis	AbrB	0.14	0.18	0.15	0.20	0.11	0.14	0.24	0.07	0.30	n/a	0.02
	AhrC	0	0.24	0.17	0.14	0.18	0.00	0.14	0.08	0.24	n/a	0.05
	AraR	0.44	0.23	0.28	0.49	0.41	0.31	0.34	0.14	0.33	n/a	0.12
	CcpA	0.47	-	0.24	0.34	0.38	0.29	0.08	0.47	0.38	n/a	0.52
	CcpC	0.39	0.00	0.22	0.00	0.05	0.00	-	0.00	-	n/a	0.00
	comA	0.27	0.24	0.19	0.05	0.17	0.00	-	0.46	-	n/a	0.07
	comK	0.36	0.02	0.12	0.19	0.14	0.07	0.02	0.04	0.08	n/a	0.00
	Fnr	1	0.98	0.39	0.07	0.00	0.00	0.52	0.53	0.54	n/a	0.81
	degU	0.05	-	0.00	0.04	0.05	0.04	0.04	0.08	0.06	n/a	0.02
	GerE	0.11	0.00	0.06	0.15	0.14	0.07	-	0.17	-	n/a	0.06
	GlnR	0.4	-	0.00	0.00	0.10	0.00	-	0.00	-	n/a	0.00
	GltC	0.4	0.16	0.28	0.00	0.32	0.00	-	0.28	-	n/a	0.00
	Hpr	0.3	0.08	0.05	0.04	0.17	0.00	-	0.00	-	n/a	0.22
	LexA	0.61	0.59	0.58	0.00	0.64	0.00	-	0.60	-	n/a	0.45
	LicT	0.16	0.04	0.10	0.05	0.09	0.15	0.00	0.20	-	n/a	0.23
	Mta	0.14	0.07	0.09	0.14	0.12	0.00	0.00	0.09	-	n/a	0.12
	PhoB	0.06	0.12	0.07	0.06	0.06	0.06	0.26	0.05	0.12	n/a	0.00
	ResD	0.11	0.19	0.11	0.09	0.05	0.00	-	0.00	0.07	n/a	0.22
	SigD	0.15	0.17	0.08	0.21	0.15	0.00	-	0.03	-	n/a	0.23
	SigW	0.19	0.52	0.15	0.32	0.21	0.01	0.54	0.60	-	n/a	0.36
	SigX	0.06	0.47	0.11	0.05	0.07	0.05	0.16	0.25	-	n/a	0.06
	Spo0A	0	0.47	0.62	0.25	0.42	0.23	0.24	0.00	-	n/a	0.00
	SpoIIID	0.04	0.05	0.06	0.10	0.17	0.08	0.15	0.14	-	n/a	0.23
	XylR	0.67	0.30	0.49	0.00	0.35	0.35	-	0.02	-	n/a	0.19
E. coli	ArcA	0.00	-	0.01	0.07	0.00	0.00	-	0.00	-	n/a	0.17
	CaiF	0.27	0.22	0.19	0.32	0.11	0.00	-	0.21	-	n/a	0.00
	Crp	0.21	0.03	0.10	0.07	0.25	0.07	0.28	0.07	0.18	n/a	0.00
	DnaA	0.02	-	0.14	0.09	0.09	0.00	-	0.32	-	n/a	0.00
	Fis	0.03	0.03	0.05	0.05	0.05	0.01	0.07	0.04	-	n/a	0.05
	Fnr	0.41	0.45	0.29	0.01	0.52	0.10	0.15	0.32	0.16	n/a	0.22
	GlpR	0.34	0.16	0.20	0.06	0.23	0.14	-	0.21	-	n/a	0.00
	IHF	0.10	0.05	0.03	0.03	0.00	0.03	0.05	0.00	-	n/a	-
	Lpr	0.13	0.00	0.09	0.00	0.11	0.00	-	0.00	-	n/a	0.00
	narL	0.06	0.01	0.20	0.15	0.25	0.09	0.21	0.11	0.03	n/a	0.05